Presentation
Agenda

Parallel Matlab Computation for STAP Clutter Scattering Function Estimation and Moving Target Estimation*

Roger Chamberlain, Daniel R. Fuhrmann, John Maschmeyer, and ***Lisandro Boggio***
Washington University, St. Louis, Missouri
Phone: 314-935-5708
Email Address: roger@ccrc.wustl.edu
Email Address: danf@ee.wustl.edu

Abstract

Conventional moving target estimation in STAP radar applications is based, in part, on adaptive clutter scattering function estimation techniques. These techniques classically rely on radar return data from adjacent range gates to estimate the clutter scattering function for the range gate of interest. Here, we are interested in using geographical information systems in conjunction with accurate platform positioning information as the basis for the clutter scattering function estimation. The goal is to improve the effectiveness of moving target estimation techniques by providing additional information to the decision-making process.

Our problem formulation is consistent with the model for space-time adaptive processing (STAP) presented in [1]. A pulse-Doppler radar platform with multiple transmit/receive elements emits several pulse train along an arbitrary flight path, such as a circle around the region of interest. Each pulse train is assumed to be perfectly coherent within one coherent processing interval (CPI), but different pulse trains are assumed noncoherent with respect to one another. The ground region is subdivided into pixels, or ground patches. The range and angle of each ground patch with respect to the platform for each transmitted pulse is assumed known, along with the illumination pattern. The received data for one pulse is modeled as the sum of the returns from all of the ground patches, each modulated by the transmit illumination. The data from all pulses or viewpoints is modeled in this way. Maximum-likelihood methodology is used to estimate the unknown scattering function [2] and a variant of the Adaptive Matched Filter [3] is used for moving target estimation.

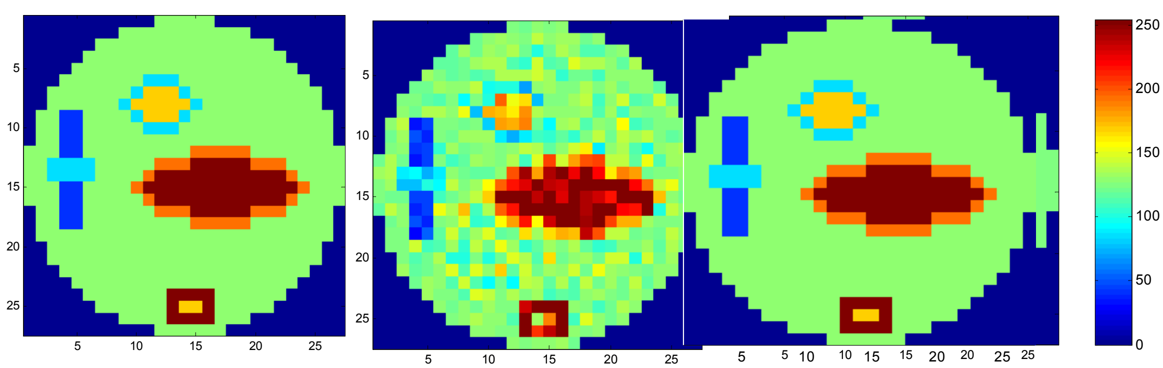


Figure 1: Artificial scattering function example. True scattering function, estimated scattering function, and estimated with land-use aggregation.

* This work is supported by DARPA through a contract with AFRL, No. F30602-03-2-0043.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **01 FEB 2005**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Parallel Matlab Computation for STAP Clutter Scattering Function Estimation and Moving Target Estimation**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Washington University, St. Louis, Missouri**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM00001742, HPEC-7 Volume 1, Proceedings of the Eighth Annual High Performance Embedded Computing (HPEC) Workshops, 28-30 September 2004 Volume 1., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**27**	

Figure 1 illustrates the scattering function estimation. This is an artificially low-dimensional simulation in which there are 550 ground patches, each associated with one of 9 different land-use values. The land-use value determines the clutter scattering function in each ground patch.
In the data collection scenario the radar platform moves in a circular flight path around the region of interest, as illustrated in Figure 2. The left panel of Figure 1 shows the true scattering function, the middle panel shows the estimated scattering function when the land use is not assumed known, and the right panel shows the estimated scattering function when pixels with the same land-use are assumed to have the same scattering function value (we call this land-use aggregration).

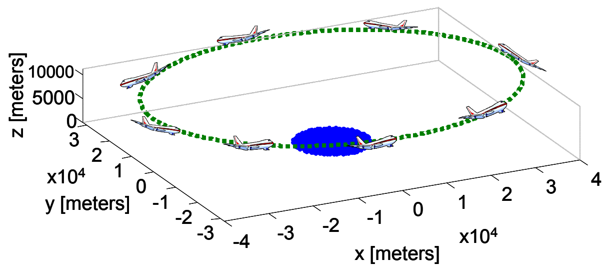


Figure 2: Simulated flight pattern for artificial scattering function example

An important question that arises is the assessment of the computational requirements associated with this application as the data set is scaled up to realistic sizes. To investigate this issue, a 15 km radius region centered on the Lake of the Ozarks in central Missouri is used as a full data set test case. With a pixel (ground patch) dimension of approximately 30 m, the region of interest comprises about 200,000 pixels. Figure 3 shows the illumination pattern on the ground for the full data set from a pair of looks. The digital terrain elevation map comes from a publicly-available website operated by the U.S. Geological Survey [4].

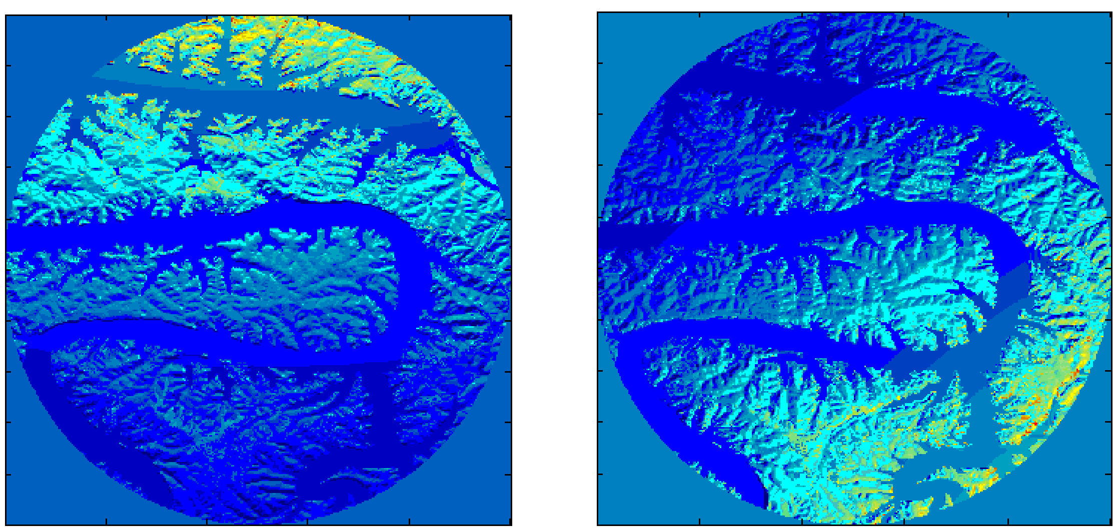


Figure 3: Illumination of full data set from different looks.

Our current activities include the execution of clutter scattering function estimation and moving target estimation on the full dataset illustrated above. The scattering function is based on 21 classes of land cover provided by the geographic information system. Given the large memory requirements for each look (greater than 250 MB/look); the computations are being executed in parallel (using MatlabMPI [5]) across 9 processors in a master-slave arrangement. The master processor is responsible for overall coordination of the computations, and the 8 slave processors are each responsible for an individual look. We will present quantitative performance measurements on these parallel computations, including parallel efficiency as well as quality of results on the clutter scattering function and moving target estimates.

[1] J. Ward. Space-time adaptive processing for airborne radar. MIT Lincoln Laboratory Technical Report 1015, December 1994.

[2] D. Fuhrmann and L. Boggio. Radar imaging from multiple viewpoints and multiple noncoherent data sets. In *Proc.2004 Conf. Information Science and Systems,* Princeton University, Princeton, NJ, March 2004.

[3] F. Robey, D. Fuhrmann, E. Kelly, and R. Nitzberg. A CFAR adaptive matched filter detector. *IEEE Trans. Aerospace and Electronic Systems*, vol. 28, no. 1, pp. 208-206, January 1992.

[4] The National Map Seamless Data Distribution System, http://seamless.usgs.gov

[5] Jeremy Kepner. Parallel Programming with MatlabMPI, In *Proc. of High Performance Embedded Computing Workshop*, September 2001.

Abstract

Agenda

Washington University in St.Louis

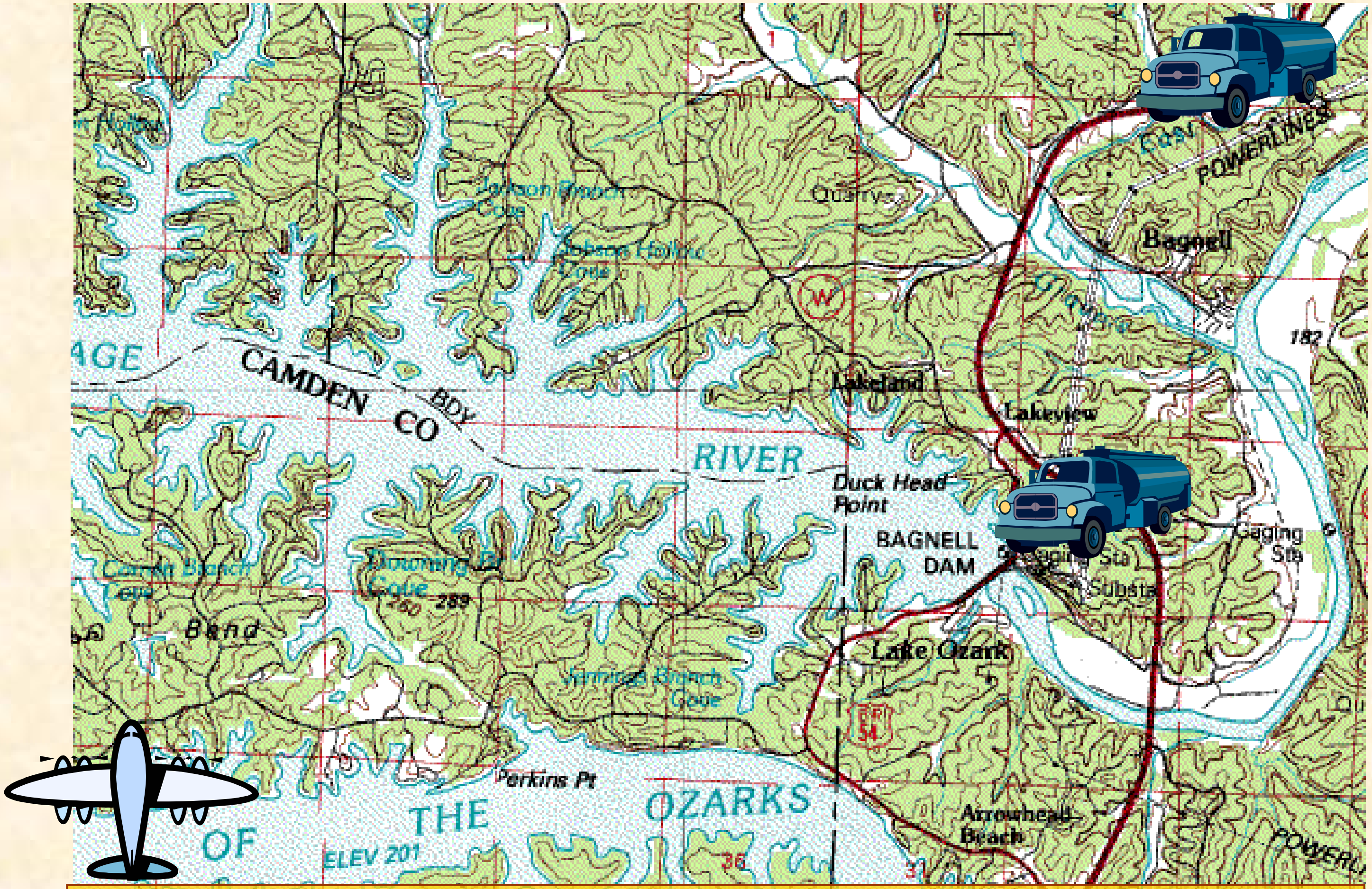
Parallel Matlab Computation for STAP Clutter Scattering Function Estimation and Moving Target Estimation

Roger Chamberlain, Daniel R. Fuhrmann, John Maschmeyer, and Lisandro Boggio

School of Engineering and Applied Science
Washington University in St. Louis

This work was supported by the U.S. Defense Advanced Research Projects Agency through a contract with the U.S. Air Force Research Laboratory, No. F30602-03-2-0043.

Context



Problem: Detect ground moving targets in the presence of ground clutter

Context

- Wide-area surveillance airborne radar
- Arbitrary flight path
- Multiple sensors and Doppler pulses
- Space-time adaptive processing (STAP)
  - Better knowledge of the clutter covariance matrix gives better detection performance

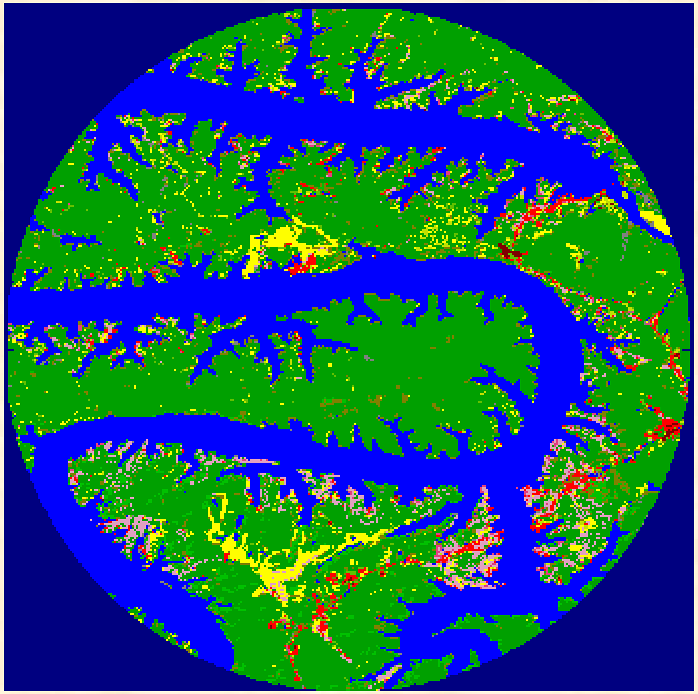
Objective: Estimate the clutter covariance matrix and detect moving targets

Approach

- Ground subdivided into pixels or ground patches
- Known range and angle of each patch with respect to airborne platform
- Known illumination pattern
- Received data: sum of returns from targets and all patches on the ground
- Prior knowledge is available:
  - Digital Terrain Elevation Maps
  - Land use information

Terrain Simulation

- **Region of Interest**
  - Lake of the Ozarks
  - 15 km diameter
  - 197,316 pixels
  - 30m resolution



Datasets

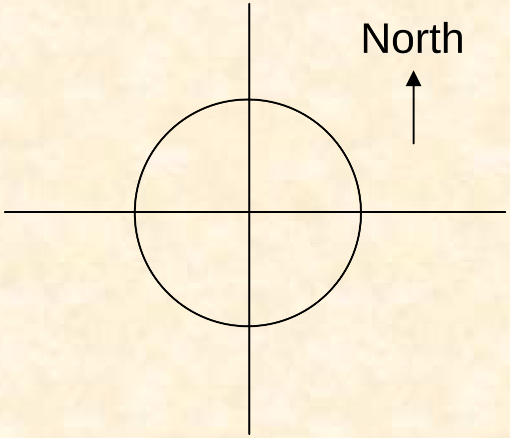
- Obtained from USGS Seamless Data Server
  - 30m resolution
- Digital Elevation Model
  - Used for modeling geometry
- Land Use
  - Scattering function based on 21 classes of land cover
    - 9 primary classes
      - Water, Developed, Barren, Forested Upland, Shrubland, Non-Natural Woody, Herbaceous Upland Natural/Semi-natural Vegetation, Herbaceous Planted/Cultivated, Wetlands
    - Each class contains one or more categories, e.g.
      - Open Water, High-Intensity Residential, Deciduous Forest, Row Crops
  - Scattering function chosen arbitrarily for simulation

Coordinate Systems

- Datasets referenced in spherical coordinates
  - Latitude, Longitude, Elevation
- Convert to Cartesian Coordinates
  - Simpler to use over small region
  - Computations can be made independent of Earth model

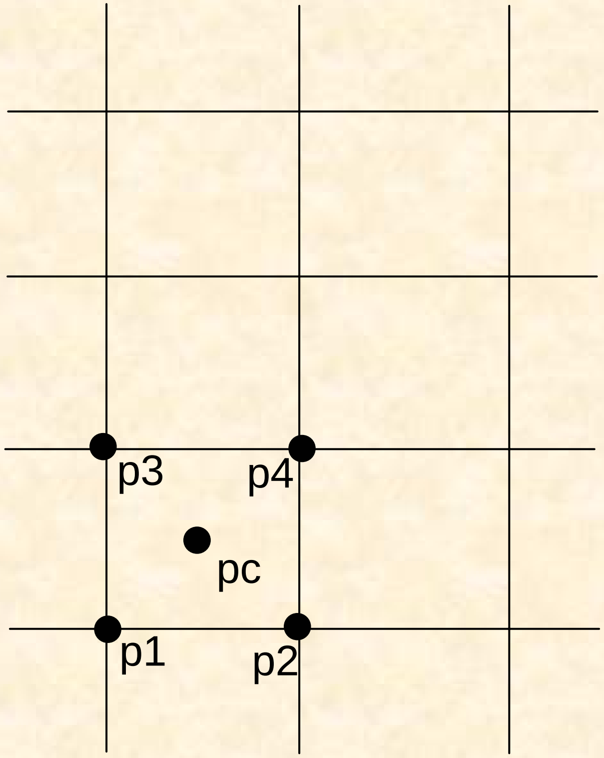
Coordinate Conversion

- First Stage
  - Origin at Earth's center
  - Use Geodetic Reference System 1980 (GRS80)
- Second Stage
  - Move origin to center of region of interest
  - Elevation along Z-axis
  - North along positive Y-axis

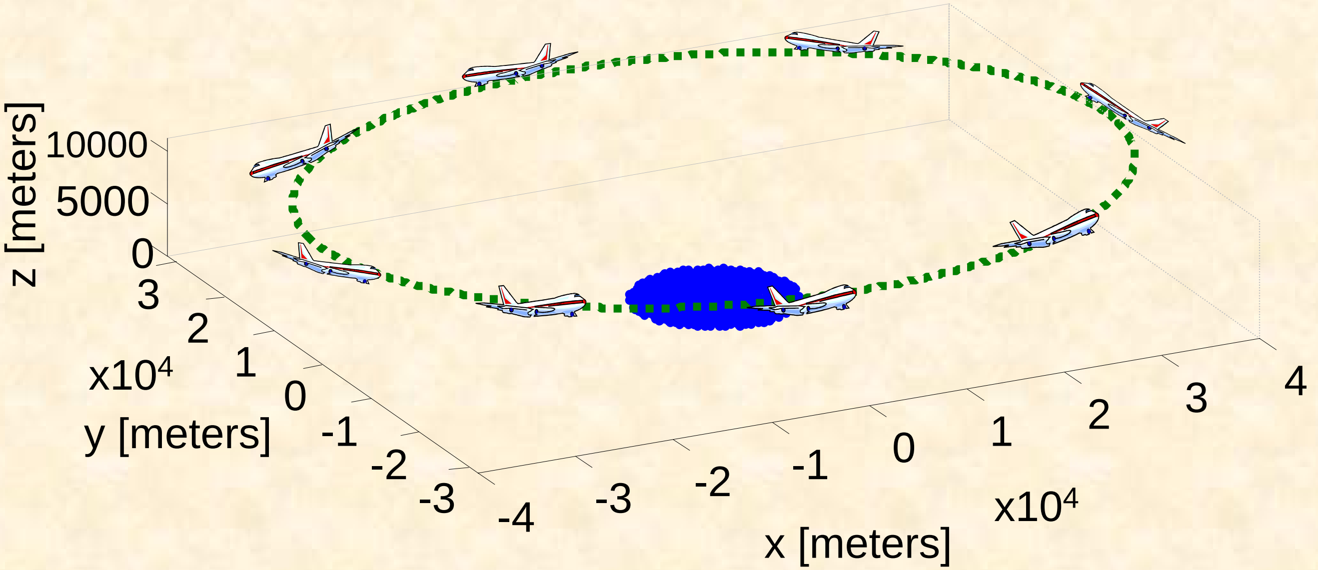


Coordinate Systems

- Adjacent data samples grouped into patches
  - Each patch, or pixel, contains:
    - Location for each corner
    - Location of center
    - Scattering function
    - Normal vectors
- 197,316 pixels in all



Simulation Setup



- Platform moves around region of interest
  - Actual flight path is arbitrary
- Eight looks

SIMULATION PARAMETERS

- Platform
  - Flies in circular path around region
  - Radius 25 km
  - Altitude 7 km
  - 8 different viewpoints
- Radar
  - f_c : 10 GHz
  - BW: 10 MHz
  - PRF: 2 KHz
  - Pulses per CPI: 38
  - ULA elements: 12
  - Range gates: 990

Geometry Parameters

- Geometry dependent parameters required for simulation
  - Range to each pixel
  - Projected area of each pixel
    - Incident energy incorporates range and projected area of patch
  - Occluded pixels
    - Patches hidden from radar are removed using Z-buffer algorithm
      - Patches sorted by distance from radar
      - Any patch facing backwards or directly behind another is removed
  - Angle between platform's velocity vector and line of sight to each pixel

Datacube Generation

- Received data from a single patch

Return from n^{th} path is a random variable

$$u_k(n) \sim \mathcal{CN}\left(0, \lambda_k(n)\sigma_n\right)$$

Radar response vector

$$\mathbf{z}_k = \mathbf{s}_k + \mathbf{n}_k + \sum_{n=1}^{N} u_k(n)\,\mathbf{a}_k(n)$$

receiver noise

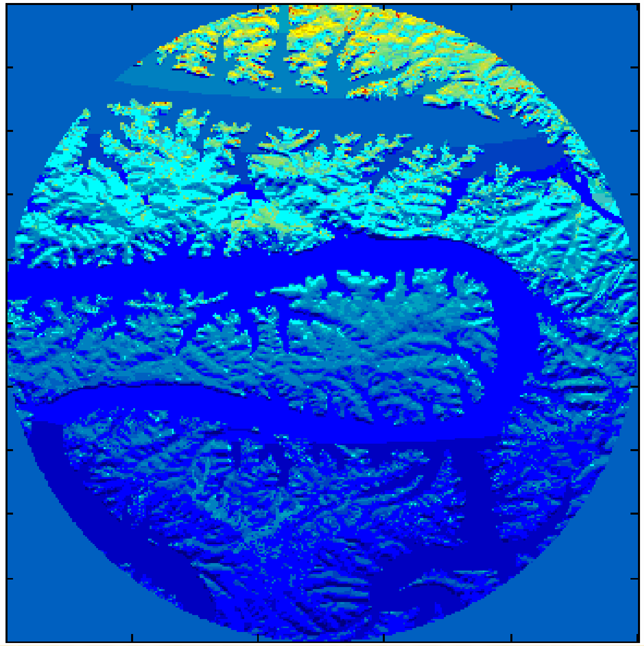
$$\mathbf{n}_k(n) \sim \mathcal{CN}\left(0, \epsilon \mathbf{I}_M\right)$$

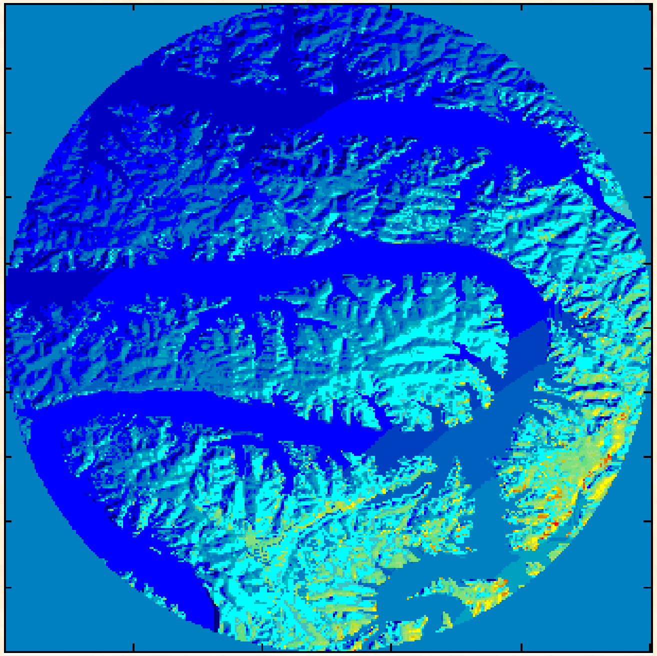
return from targets

- Response at a single range gate
  - Sum over all patches in range gate

ILLUMINATION

Illumination from different looks





Scattering Function Estimation

- Prototype designed and tested first
  - Implements EM algorithm
  - Uses a Small-Scale Dataset with 554 pixels
- EM algorithm requires response vector for each pixel, in each look
  - For Small-Scale Simulation
    - 2,020,992 complex doubles
    - 30.8 MB of data
- Large-Scale Simulation contains 197,316 pixels
  - 719,808,768 complex doubles
  - 10.73 GB

Memory Reducing Techniques

- Maintain only the Doppler and spatial vectors
  - Compute Kronecker product as needed
  - Reduces requirements to 1.17 GB

- Lookup Table
  - Finely sampled table containing Doppler and spatial vectors
  - Indexed by a single value
  - Further reduces memory requirements
    - 10.64 MB when using a 10,000 entry table

The Need For Parallelism

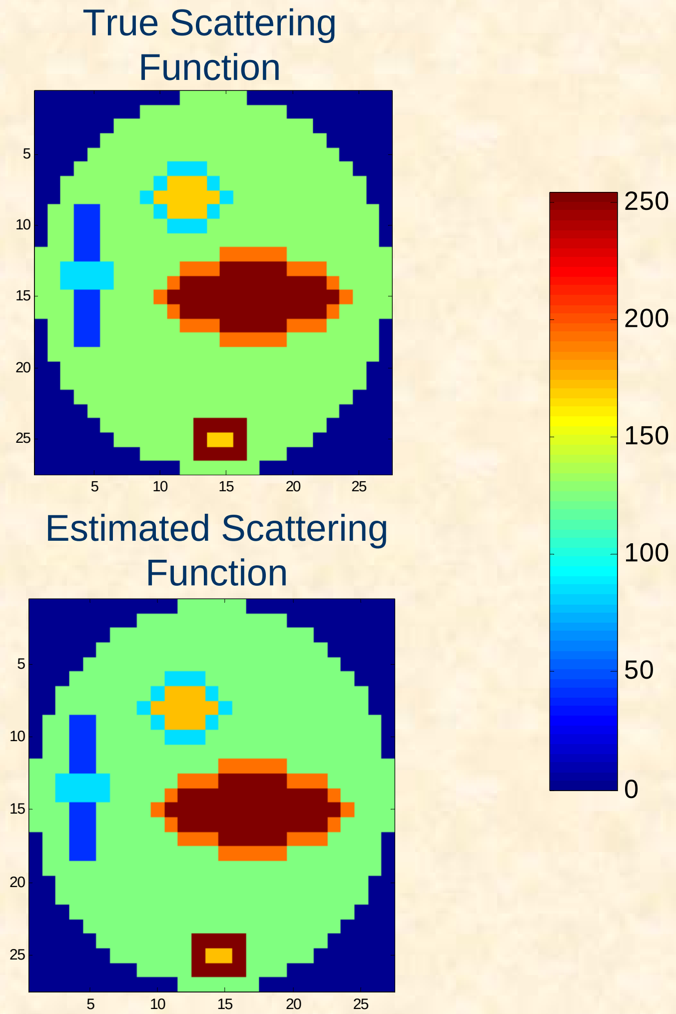
- The EM Algorithm can be parallelized in multiple ways
  - Across looks
  - Across range gates
- Parallelism improves the algorithm
  - Significant speedup in processing time
  - Additional physical memory available
    - Only 150 MB needed per look for the response vectors (250 MB when all other necessary data are included)
  - More effective cache
    - Possible gain when using a Lookup Table

Parallelism Using MatlabMPI

- MatlabMPI provides parallel interface
  - Allows passing of messages between multiple systems that share a file system
- Use 9 parallel threads (1 master, 8 slaves)
  - Slaves perform iterations of the EM algorithm on a single look
  - Master provides slaves with data and collects results from each iteration
- Messages only sent at beginning and end of each iteration

Results

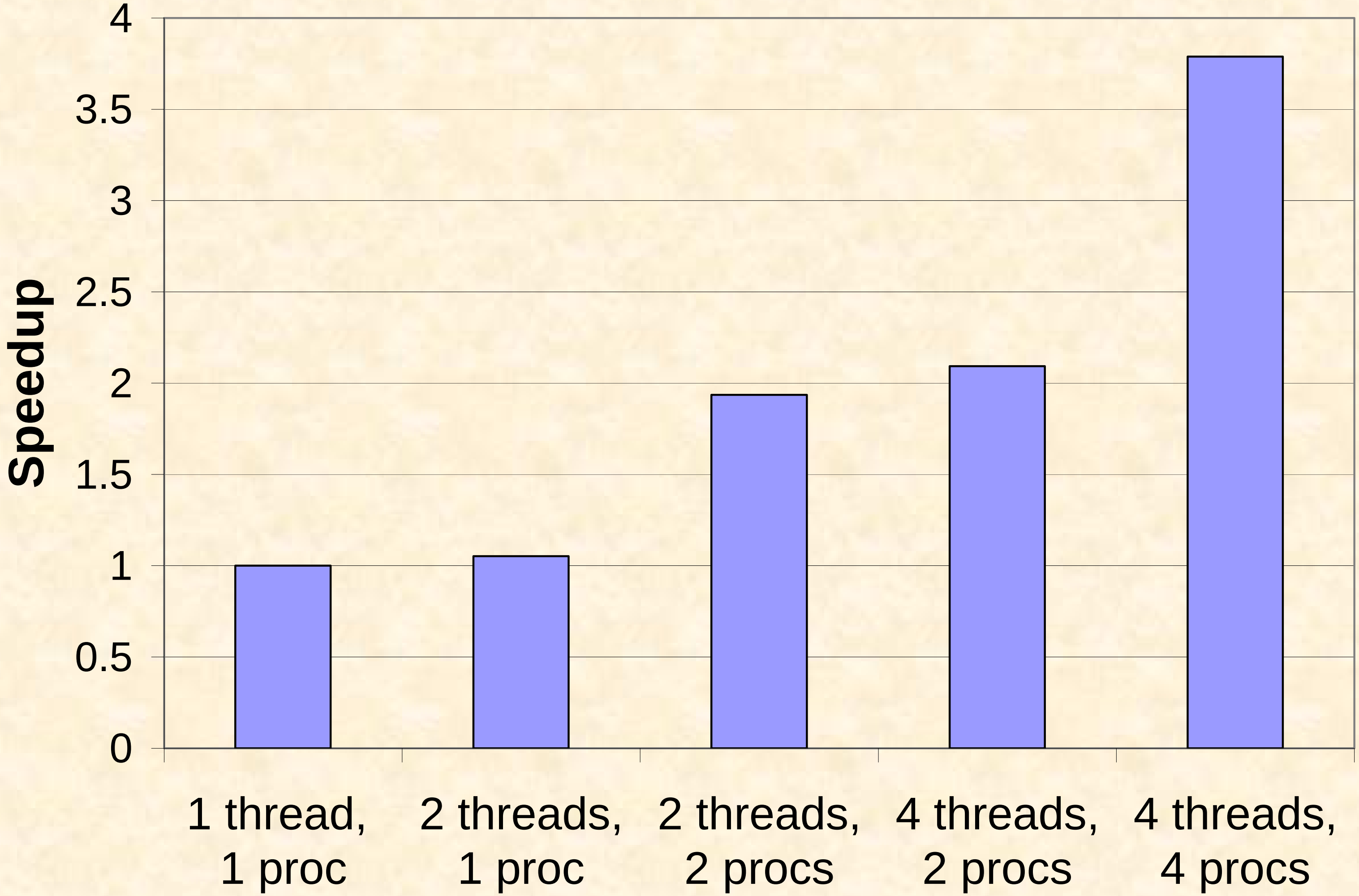
- Small-Scale Simulation
  - Provides results identical to prototype version
  - Runs 4% slower than non-parallel version
    - Computation for a single look is too fast to gain from parallelism
    - Message passing overhead too large
    - Not a problem for full-scale simulation



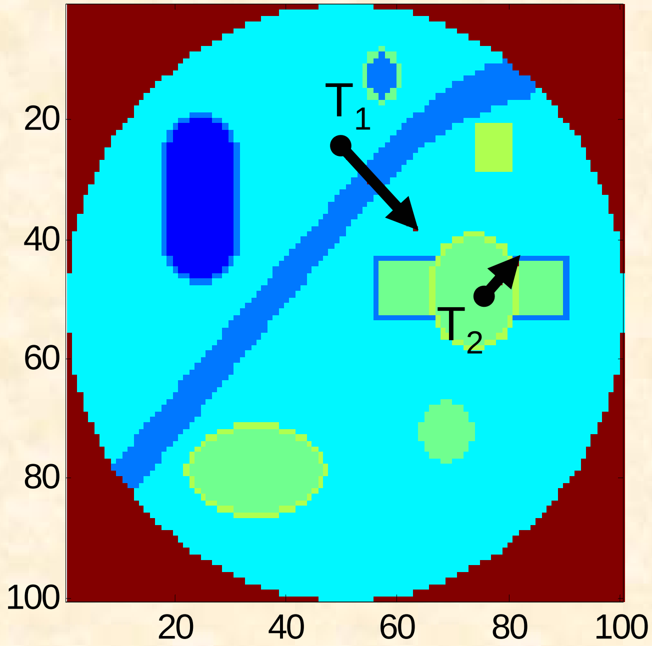
Results

- Full-Scale Simulation
  - Does not use Lookup Table
  - To avoid large messages, some inputs are read from disk
- Execution Environment
  - 2.4 GHz Pentium IV processors w/ hyperthreading
  - 1 GB RAM each
  - 4 nodes

Results


Speedup
4
3.5
3
2.5
2
1.5
1
0.5
0
1 thread, 1 proc
2 threads, 1 proc
2 threads, 2 procs
4 threads, 2 procs
4 threads, 4 procs

Detection Example

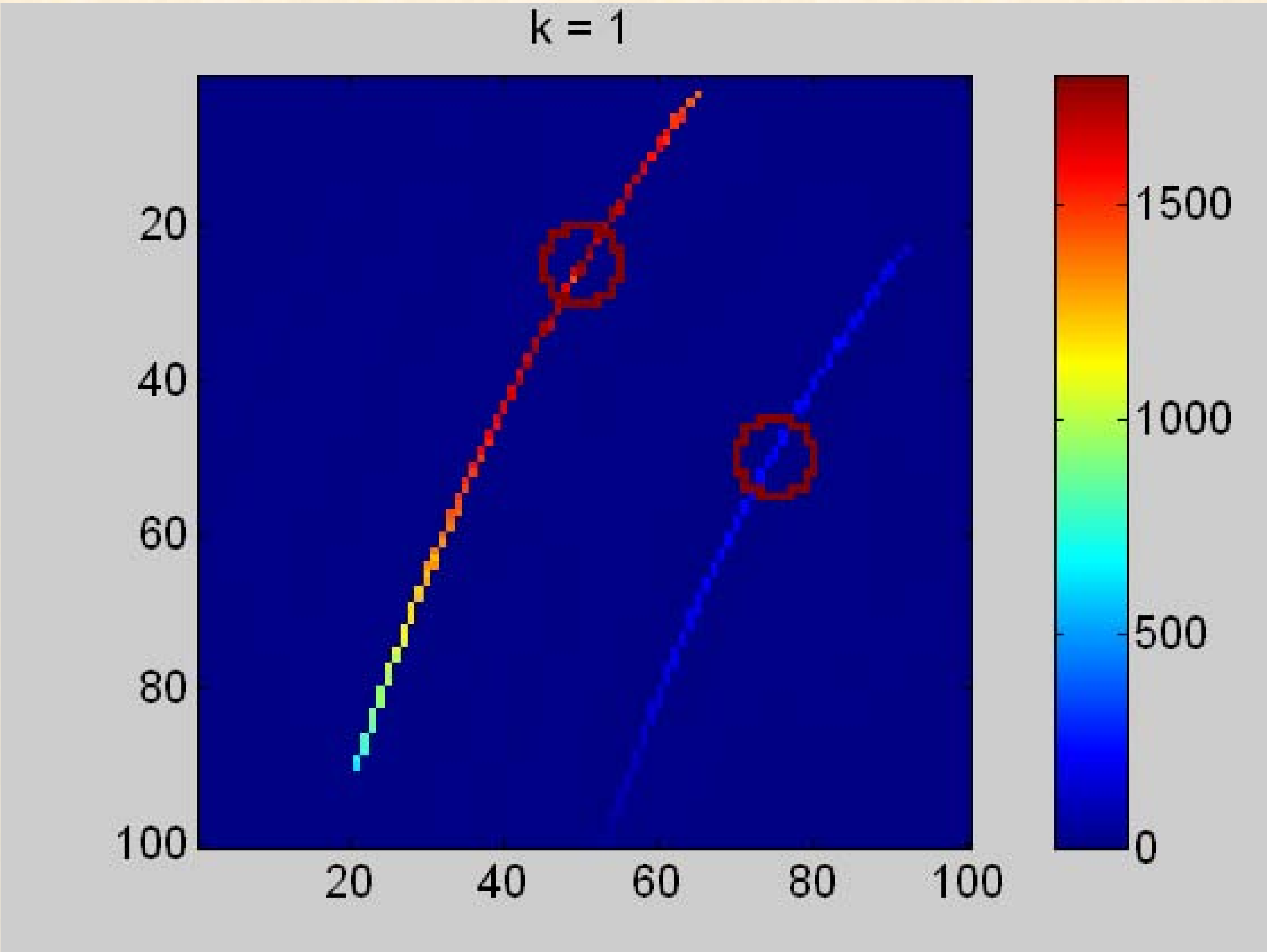


- Two artificial targets
- In small-scale environment
- Binary detection problem

$$\boldsymbol{z}_k = \begin{cases} \boldsymbol{s}_k + \boldsymbol{A}_k \boldsymbol{u}_k + \boldsymbol{n}_k & H_1 \\ \boldsymbol{A}_k \boldsymbol{u}_k + \boldsymbol{n}_k & H_0 \end{cases}$$

- Use adaptive matched filter

Adaptive Matched Filter Detector



Current and Future Work

- Completing the Full-Scale Simulation
  - Long runtimes are still a problem
- Moving Target Estimation on Full-Scale System